

リフレクションに基づくエージェントプログラミング 支援機構について

1H-10

副島 大和 大園 忠親 新谷 虎松
名古屋工業大学知能情報システム学科

1. はじめに

計算機科学において、エージェントは、自律したソフトウェアとして扱われている。知的で自律したエージェントから構成されるマルチエージェントシステムを容易に構築することを可能にするには、リフレクション[1]機構の実装が有効である。知的で自律した拡張性の高いエージェントを実現するための言語として、リフレクティブな制約論理型言語 R X F [2]がある。

R X Fは、階層的なアーキテクチャに基づくエージェントの実装をサポートする。エージェント実装のための階層的なアーキテクチャは、刻々と変化する動的環境に対応可能な機構の実現のために提案された。実世界を扱うエージェントが知的で自律的に動作するために、階層的なアーキテクチャを利用することは大変に有効である。本研究では、階層的なアーキテクチャに基づくエージェントプログラミングにおいて、プログラマを補助する機構を実装した。本研究では、階層的なアーキテクチャをもつエージェントプログラミングを補助する仕組みとして、エージェントの階層構造の視覚化を行う。本研究では、視覚化を実現するために、コンポーネントプログラミングの技術を利用する。

2. R X Fにおける階層的アーキテクチャとリフレクション

R X Fにおける階層的アーキテクチャは、各階層がスレッドにより構成される。各階層は他の階層と相対的にメタやベースといった関係をもつ。メタレベルの階層は、ベースレベルの階層を参照・改変することが可能である。ベースレベルの階層はメタレベルの階層に何らかの実行を依頼することが可能である。メタレベルの階層にベースレベルの階層が、自分自身（ベースレベルの階層）に関する参照・改変を依頼することによって、内省や自己改変といったリフレクションの機能を実現している。エージェントは、このような階層構造をもつ。

図1が、R X Fにおけるエージェントの概念図である。図1では、丸がエージェントを表

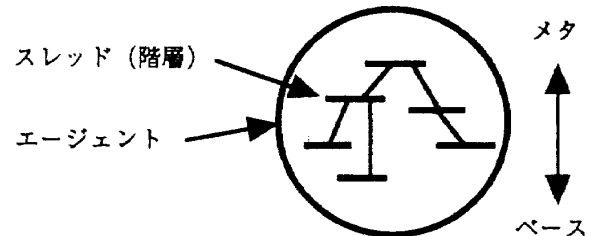


図1 エージェントの概念図

す。横線が一つのスレッド、各スレッドを結ぶ斜め線が各スレッドがメタやベースという関係を持つことを表している。図において、上に配置されている程メタなレベルを表している。

3. 問題点と解決法

階層的なアーキテクチャを実装するためのエージェントプログラミングにおいて、階層の数が多くなると、ソースを概観しただけでは、エージェントの階層構造を理解することが困難になる。この問題点の軽減のために、エージェントの階層構造を視覚化する。図1が本視覚化の例であり、エージェントの階層構造が、ソースを直接見るよりは、直感的に分かり易くなる。

4. 視覚化の実現とコンポーネント

本研究では、コンポーネントプログラミングに基づく視覚化を実現した。エージェントの階層構造を構成するスレッドを、部品化して部品をマウス操作で組み合わせることによって、簡単に階層構造を実現できる。コンポーネントプログラミングをおこなうことで、部品が階層構造の概観を構成するので、視覚化が実現できる。

5. 機能と実装

コンポーネントプログラミングを実現するために、ツールボックスと呼ばれるコンポーネントライブラリを一覧表示させるランチャを作成した。ツールボックスを用いた開発は、ツールボックスからドラッグ&ドロップすることによって達成される。これにより、エージェントを直感的に実装することが可能となった。また、エージェントにスレッドやデータベースを容易に追加できるようなテンプレートを生成させる機能も同時に実装している。スレッド間の

An Agent Programming Support System Based on Reflection

Hirokazu Soejima, Tadachika Ozono, Tramatsu Shintani
Dept. of Intelligence and Computer Science,
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555

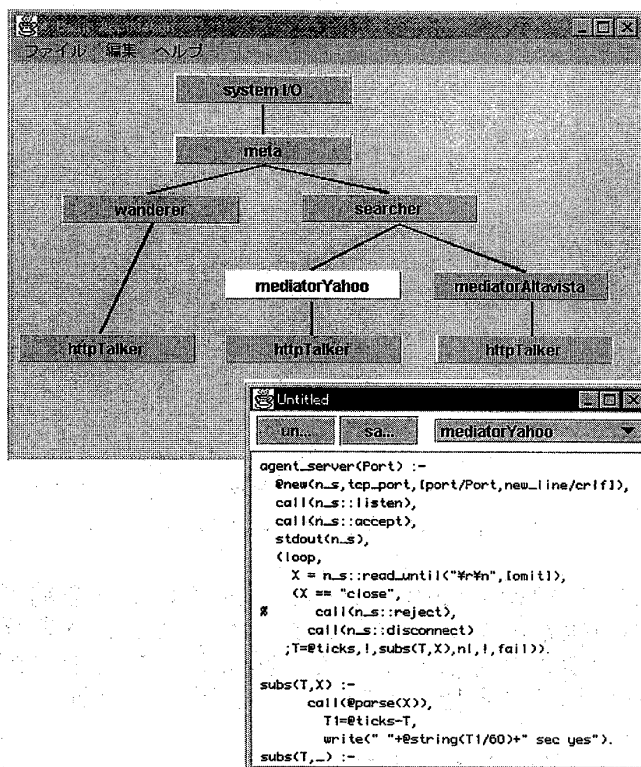


図2 インタフェースの例

階層関係をマウスで関連付けることを可能にする機能も実装している。あるスレッド上で実行中のプログラムや、データベースなどを、記述・確認・改変することを容易にする機能を実装した。

スレッドの状態を観察するために、実行中に割り込みをかけて、リフレクションを使う。割り込みにはRXFのアテンションハンドラという割り込みの機能とRXFのリフレクションの機能を利用する。実行中にある階層のスレッドをダブルクリックすると、アテンションハンドラが割り込みをかけて実行を停止する。このとき、スレッドの述語の定義（プログラム）も見ることができる。停止している際に、リフレクションを使えば、スレッドの内部状態を知ることが可能となる。

6. 実行例

上の図2が支援機構のインタフェースの例である。二つあるウィンドウのうち、左上がエージェントの内部構成をあらわしているウィンドウ、右下がスレッド（階層）などのエディタである。左上のウィンドウ一つで、エージェント一つをあらわす。エージェントをあらわすウィンドウの内部、横に長い四角が一つのスレッド（階層）をあらわしている。各スレッド同士を結ぶものが、各スレッドのメタとベースの対

応関係をあらわすものである。ウィンドウのほぼ中央で、白くなっているスレッドは、現在プログラマが注目しているスレッドをあらわす。注目しているとは、エディタでそのスレッドを編集集中であるか、実行中に、状態の観察をおこなっているということである。エディタは、スレッドやデータベースの編集と、それらをconsultやreconsultする機能をもつ。

7. まとめ

本研究によって、エージェントプログラミング支援環境を示した。階層構造をもつエージェントプログラミングにおいては、(1)階層関係の把握と階層関係に関するプログラミング、(2)スレッドの状態の把握とスレッドプログラミング、が困難である。そこで、階層関係とスレッドの状態を視覚化することによってプログラマのエージェントのプログラミングを補助する機構を実装した。また、階層構造とスレッドのプログラムをプログラマが記述することを補助するために、スレッドやスレッド同士の関係をコンポーネントプログラミングできるように、部品化を行った。コンポーネントを組み合わせることによって、プログラミングに必要な部分について、テンプレートが生成される。本研究によって、プログラマのエージェントプログラミングにおける開発の負担軽減に繋がったと考えられる。

今後の課題としては、リフレクティブなマルチエージェントシステムも視野に入れた開発環境[3]に発展させることがある。リフレクティブなマルチエージェントシステムとは、複数のエージェント同士が階層関係を構成しているエージェントシステムを指す。また、コンポーネントとテンプレートの種類を充実させることがある。

参考文献

- [1] 渡部卓雄：リフレクション、コンピュータソフトウェア、Vol. 11, No. 3, pp. 5-14 (1994)。
- [2] 大園忠親，新谷虎松：自律的エージェントのための制約論理型言語RXFにおけるリフレクション機構の設計とその実装，情報処理学会論文誌，Vol.38, No.7, pp.1361-1369 (1997)。
- [3] 副島大和，大園忠親，新谷虎松：エージェント記述言語RXFにおけるエージェントプログラミング支援機構について，人工知能学会全国大会（第12回）論文集，人工知能学会，pp.203-204, (1998)。